

Antipatterns Managing Software Organizations And People Second Edition Applied Software Engineering Series

AntiPatterns: Identification, Refactoring, and Management catalogs 48 bad management practices and environments common to software development, IT, and other organizations. The authors cover antipatterns of management, along with environmental/cultural antipatterns and personality antipatterns/phenotypes. Through the classification of these harmful practices, you will be able to correctly identify problems in your own work environment, and take action to correct them. The authors apply their extensive work and consultative experience, as well as the experience of the many professionals that they have known. This approach leads to a realistic treatment of antipattern concepts. Written for a wide audience of practitioners, the authors avoid a scholarly style, instead infusing the text with entertaining “gadgets,” including rambunctious and ribald sidebars, cartoons, stories, and jokes, as well as names for their antipatterns that are at once visual, iconic, humorous, and memorable. Following introductory material describing some management theory and how humans behave individually and in groups, the text provides the catalog of management and environmental antipatterns. The book then offers general advice on overcoming bad practices through successful interaction with clients, customers, peers, supervisors, and subordinates.

Systems Engineering and Management for Sustainable Development is a component of Encyclopedia of Technology, Information, and Systems Management Resources in the global Encyclopedia of Life Support Systems (EOLSS), which is an integrated compendium of twenty one Encyclopedias. This theme discusses: basic principles of systems engineering and management for sustainable development, including: cost effectiveness assessment; decision assessment, tradeoffs, conflict resolution and negotiation; research and development policy; industrial ecology; and risk management strategies for sustainability. The emphasis throughout will be upon the development of appropriate life-cycles for processes that assist in the attainment of sustainable development, and in the use of appropriate policies and systems management approaches to ensure successful application of these processes. The general objectives of these chapters is to illustrate the way in which one specific issue, such as the need to bring about sustainable development, necessarily grows in scope such that it becomes only feasible to consider the engineering and architecting of appropriate systems when the specific issue is imbedded into a wealth of other issues. The discussions provide an illustration of the many attributes and needs associated with the important task of utilizing information and knowledge, enabled through systems engineering and management, to engineer systems involving humans, organizations, and technology, in the support of sustainability. These two volumes are aimed at the following five major target audiences: University and College students Educators,

Professional practitioners, Research personnel and Policy analysts, managers, and decision makers and NGOs.

Software Engineering for Image Processing Systems creates a modern engineering framework for the specification, design, coding, testing, and maintenance of image processing software and systems. The text is designed to benefit not only software engineers, but also workers with backgrounds in mathematics, the physical sciences, and other engineering

Antipatterns Managing Software Organizations and People, Second Edition CRC Press

This volume examines proven software configuration management strategies to allow professionals to deliver quality software systems with the least amount of wasted effort. It is designed to help managers build and foster a development environment focused on producing optimal teamwork.

The book covers the recent new advances in software engineering and knowledge engineering. It is intended as a supplement to the two-volume handbook of software engineering and knowledge engineering. The editor and authors are well-known international experts in their respective fields of expertise. Each chapter in the book is entirely self-contained and gives in-depth information on a specific topic of current interest. This book will be a useful desktop companion for both practitioners and students of software engineering and knowledge engineering.

Scott Ambler, award-winning author of Building Object Applications that Work, Process Patterns, and More

Process Patterns, has revised his acclaimed first book, The Object Primer. Long prized in its original edition by both students and professionals as the best introduction to object-oriented technology, this book has all modeling notation rewritten in UML 2.0. All chapters have been revised to take advantage of Agile Modeling (AM), which is presented in the new chapter 2 along with other important modeling techniques. Review questions at the end of each chapter allow readers to test their newly acquired knowledge. In addition, the author takes time to reflect on the lessons learned over the past few years by discussing the proven benefits and drawbacks of the technology. This is the perfect book for any software development professional or student seeking an introduction to the concepts and terminology of object technology.

As the population ages and healthcare costs continue to soar, the focus of the nation and the healthcare industry turns to reducing costs and making the delivery process more efficient. Demonstrating how improvements in information systems can lead to improved patient care, Information and Communication Technologies in Healthcare explains how to cr

Programming has become a significant part of connecting theoretical development and scientific application computation. Computer programs and processes that take into account the goals and needs of the user meet with the greatest success, so it behooves software engineers to consider the human element inherent in every line of code they write. Research Anthology on Recent Trends, Tools, and Implications of Computer Programming is a vital reference source that examines the latest scholarly material on trends,

techniques, and uses of various programming applications and examines the benefits and challenges of these computational developments. Highlighting a range of topics such as coding standards, software engineering, and computer systems development, this multi-volume book is ideally designed for programmers, computer scientists, software developers, analysts, security experts, IoT software programmers, computer and software engineers, students, professionals, and researchers.

Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler's Children provides an independent examination of developments in Enterprise Resource Planning for Information. Major companies, research firms, and vendors are offering Enterprise Resource Planning for Information Technology, which they label as ERP for IT, IT Resource Planning and related terms. This book presents on-the-ground coverage of enabling IT governance in architectural detail, which can be used to define a strategy for immediate execution. It fills the gap between high-level guidance on IT governance and detailed discussions about specific vendor technologies. It provides a unique value chain approach to integrating the COBIT, ITIL, and CMM frameworks into a coherent, unified whole. It presents a field-tested, detailed conceptual information model with definitions and usage scenarios, mapped to both process and system architectures. This book is recommended for practitioners and managers engaged in IT support in large companies, particularly those who are information architects, enterprise architects, senior software engineers, program/project managers, and IT managers/directors.

Introduction to patterns and antipatterns. The nature of a patterns: a brief tutorial. The lost disciplines: a system engineering perspective. The father of all management

antipatterns. software engineering antipatterns and patterns.
Software configuration management pattern and antipatterns.
Management and process patterns and antipatterns.
Requirements and testing patterns and antipatterns.
Conclusions and resources.

Presents patterns for the Deliver and the Maintain and Support phases of large-scale system development.
"This book provides an overall view of the emerging field of complex data processing, highlighting the similarities between the different data, issues and approaches"--Provided by publisher.

"This book provides various aspects of intelligent information technologies as they are applied to organizations to assist in improving productivity through the use of autonomous decision-making systems"--Provided by publisher.
Information is considered both an essential element of organizational design and an asset to be processed and managed. Further research on and application of topics relating to the architecture, management, and use of information is imperative to organizational success. The Handbook of Research on Information Architecture and Management in Modern Organizations focuses on information as an essential element of organizational design and emphasizes the strategic role of knowledge transfer and management in organizations across industries. Taking a cross-disciplinary approach to information architecture and management, this publication draws on research essential to diverse organizations and is designed for use by business professionals, researchers, academicians, and upper-level students. This comprehensive reference work features key research and concepts on topics related to information functionality, information modeling, information overload, information retrieval, innovation management, organizational architecture, informed governance, and relevant applications

across industries.

Spillover of Adsorbed Species: International Symposium Proceedings

Organizational Learning and Knowledge: Concepts, Methodologies, Tools and Applications demonstrates exhaustively the many applications, issues, and techniques applied to the science of recording, categorizing, using and learning from the experiences and expertise acquired by the modern organization. A much needed collection, this multi-volume reference presents the theoretical foundations, research results, practical case studies, and future trends to both inform the decisions facing today's organizations and the establish fruitful organizational practices for the future. Practitioners, researchers, and academics involved in leading organizations of all types will find useful, grounded resources for navigating the ever-changing organizational landscape.

On behalf of the PROFES organizing committee we would like to welcome you to the 4th International Conference on Product Focused Software Process Improvement (PROFES 2002) in Rovaniemi, Finland. The conference was held on the Arctic Circle in exotic Lapland under the Northern Lights just before Christmas time, when Kaamos (the polar night is known in Finnish as "Kaamos") shows its best characteristics. PROFES has established itself as one of the recognized international process improvement conferences. Despite the current economic downturn, PROFES has attracted a record number of submissions. A total of 70 full papers were submitted and the program committee had a

difficult task in selecting the best papers to be presented at the conference. The main theme of PROFES is professional software process improvement (SPI) motivated by product and service quality needs. SPI is facilitated by software process assessment, software measurement, process modeling, and technology transfer. It has become a practical tool for quality software engineering and management. The conference addresses both the solutions found in practice and the relevant research results from academia.

Presents an approach to software architecture that takes organizational issues into consideration. The approach uses a series of five principles--vision, rhythm, anticipation, partnering, and simplification--to reveal hidden risks and opportunities of software architecture. Complementing these principles are criteria, patterns, and antipatterns. The criteria help assess how well each principle is being performed currently, and the patterns and antipatterns provide guidance on how to apply the principles. c. Book News Inc.

"The AntiPatterns authors have clearly been there and done that when it comes to managing software development efforts. I resonated with one insight after another, having witnessed too many wayward projects myself. The experience in this book is palpable." -John Vlissides, IBM Research "This book allows managers, architects, and developers to learn from the painful mistakes of others. The high-level AntiPatterns on software architecture are a particularly valuable contribution to software engineering. Highly recommended!" -Kyle Brown Author of The Design

Patterns Smalltalk Companion "AntiPatterns continues the trend started in Design Patterns. The authors have discovered and named common problem situations resulting from poor management or architecture control, mistakes which most experienced practitioners will recognize. Should you find yourself with one of the AntiPatterns, they even provide some clues on how to get yourself out of the situation." -Gerard Meszaros, Chief Architect, Object Systems Group

Are you headed into the software development mine field? Follow someone if you can, but if you're on your own-better get the map! AntiPatterns is the map. This book helps you navigate through today's dangerous software development projects. Just look at the statistics: * Nearly one-third of all software projects are cancelled. * Two-thirds of all software projects encounter cost overruns in excess of 200%. * Over 80% of all software projects are deemed failures. While patterns help you to identify and implement procedures, designs, and codes that work, AntiPatterns do the exact opposite; they let you zero-in on the development detonators, architectural tripwires, and personality booby traps that can spell doom for your project. Written by an all-star team of object-oriented systems developers, AntiPatterns identifies 40 of the most common AntiPatterns in the areas of software development, architecture, and project management. The authors then show you how to detect and defuse AntiPatterns as well as supply refactored solutions for each AntiPattern presented.

Modern-day projects require software and systems engineers to work together in realizing architectures of

large and complex software-intensive systems. To date, the two have used their own tools and methods to deal with similar issues when it comes to the requirements, design, testing, maintenance, and evolution of these architectures. *Software and Systems Architecture in Action* explores practices that can be helpful in the development of architectures of large-scale systems in which software is a major component. Examining the synergies that exist between the disciplines of software and systems engineering, it presents concepts, techniques, and methods for creating and documenting architectures. The book describes an approach to architecture design that is driven from systemic quality attributes determined from both the business and technical goals of the system, rather than just its functional requirements. This architecture-centric design approach utilizes analytically derived patterns and tactics for quality attributes that inform the architect's design choices and help shape the architecture of a given system. The book includes coverage of techniques used to assess the impact of architecture-centric design on the structural complexity of a system. After reading the book, you will understand how to create architectures of systems and assess their ability to meet the business goals of your organization. Ideal for anyone involved with large and complex software-intensive systems, the book details powerful methods for engaging the software and systems engineers on your team. The book is also suitable for use in undergraduate and graduate-level courses on software and systems architecture as it exposes students to the concepts and techniques used

to create and manage architectures of software-intensive systems.

?Software is continuously increasing in complexity. Paradigmatic shifts and new development frameworks make it easier to implement software – but not to test it. Software testing remains to be a topic with many open questions with regard to both technical low-level aspects and to the organizational embedding of testing. However, a desired level of software quality cannot be achieved by either choosing a technical procedure or by optimizing testing processes. In fact, it requires a holistic approach. This Brief summarizes the current knowledge of software testing and introduces three current research approaches. The base of knowledge is presented comprehensively in scope but concise in length; thereby the volume can be used as a reference. Research is highlighted from different points of view. Firstly, progress on developing a tool for automated test case generation (TCG) based on a program's structure is introduced. Secondly, results from a project with industry partners on testing best practices are highlighted. Thirdly, embedding testing into e-assessment of programming exercises is described.

By bringing together various current directions, Software Project Management in a Changing World focuses on how people and organizations can make their processes more change-adaptive. The selected chapters closely correspond to the project management knowledge areas introduced by the Project Management Body of Knowledge, including its extension for managing software projects. The contributions are grouped into

four parts, preceded by a general introduction. Part I “Fundamentals” provides in-depth insights into fundamental topics including resource allocation, cost estimation and risk management. Part II “Supporting Areas” presents recent experiences and results related to the management of quality systems, knowledge, product portfolios and global and virtual software teams. Part III “New Paradigms” details new and evolving software-development practices including agile, distributed and open and inner-source development. Finally, Part IV “Emerging Techniques” introduces search-based techniques, social media, software process simulation and the efficient use of empirical data and their effects on software-management practices. This book will attract readers from both academia and practice with its excellent balance between new findings and experience of their usage in new contexts. Whenever appropriate, the presentation is based on evidence from empirical evaluation of the proposed approaches. For researchers and graduate students, it presents some of the latest methods and techniques to accommodate new challenges facing the discipline. For professionals, it serves as a source of inspiration for refining their project-management skills in new areas. Software development has been a troubling since it first started. There are seven chronic problems that have plagued it from the beginning: Incomplete and ambiguous user requirements that grow by $>2\%$ per month. Major cost and schedule overruns for large applications $> 35\%$ higher than planned. Low defect removal efficiency (DRE) Cancelled projects that are not

completed: > 30% above 10,000 function points. Poor quality and low reliability after the software is delivered: > 5 bugs per FP. Breach of contract litigation against software outsource vendors. Expensive maintenance and enhancement costs after delivery. These are endemic problems for software executives, software engineers and software customers but they are not insurmountable. In *Software Development Patterns and Antipatterns*, software engineering and metrics pioneer Capers Jones presents technical solutions for all seven. The solutions involve moving from harmful patterns of software development to effective patterns of software development. The first section of the book examines common software development problems that have been observed in many companies and government agencies. The data on the problems comes from consulting studies, breach of contract lawsuits, and the literature on major software failures. This section considers the factors involved with cost overruns, schedule delays, canceled projects, poor quality, and expensive maintenance after deployment. The second section shows patterns that lead to software success. The data comes from actual companies. The section's first chapter on Corporate Software Risk Reduction in a Fortune 500 company was based on a major telecom company whose CEO was troubled by repeated software failures. The other chapters in this section deal with methods of achieving excellence, as well as measures that can prove excellence to C-level executives, and with continuing excellence through the maintenance cycle as well as for software development.

Written by one of the best known object-oriented practitioners in the business, Process Patterns is based on proven, real-world techniques. Scott Ambler shows readers how to successfully deliver large-scale applications using object technology and carefully describes how one develops applications that are truly easy to maintain and to enhance. He shows how such projects can be supported and points out what is necessary to ensure that one's development efforts are of the best quality. His object-oriented software process (OOSP) is geared toward medium to large-size organizations that need to internally develop software to support their main line of business. Developers and project managers who have just taken their first OO development course will find this book essential. It describes the only OOSP to take the true needs of development into consideration, including cross-project, maintenance, operations, and support issues. This book uses the Unified Modeling Language (UML). This is a detailed summary of research on design rationale providing researchers in software engineering with an excellent overview of the subject. Professional software engineers will find many examples, resources and incentives to enhance their ability to make decisions during all phases of the software lifecycle. Software engineering is still primarily a human-based activity and rationale management is concerned with making

design and development decisions explicit to all stakeholders involved.

Operations Anti-Patterns, DevOps Solutions shows how to implement DevOps techniques in the kind of imperfect environments most developers work in.

Part technology tutorial, part reference manual, and part psychology handbook, this practical guide shows you realistic ways to bring DevOps to your team when you don't have the flexibility to make sweeping changes in organizational structure.

Summary Operations Anti-Patterns, DevOps Solutions shows how to implement DevOps techniques in the kind of imperfect environments most developers work in. Part technology tutorial, part reference manual, and part psychology handbook, this practical guide shows you realistic ways to bring DevOps to your team when you don't have the flexibility to make sweeping changes in organizational structure. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology To some extent, all organizations—even yours—suffer from poor development practices, garbled communications, and outdated legacy systems. The good news is DevOps can help you improve your processes. First, however, you'll need to recognize the core issues holding you back. This book empowers you to deliver DevOps with limited resources while navigating the office politics and

entrenched mindsets that are all too common in actual workplaces. About the book Operations Anti-Patterns, DevOps Solutions offers clear steps for transforming development and communication. Using jargon-free language, this book describes incremental techniques that pay off immediately. Streamline your workflow, manage unplanned time, and build operational metrics. Whatever your issues, this book holds the keys to organizational success. What's inside Turn failure into opportunity Drive change through culture Break down knowledge silos Settle middle management turf wars About the reader For team leaders and managers. About the author Jeffery D. Smith has been in the technology industry for over 15 years. He has managed DevOps transformations at the ad-tech firm Centro and the online ordering platform Grubhub. Table of Contents 1 The DevOps ingredients 2 The paternalist syndrome 3 Operational blindness 4 Data instead of information 5 Quality as a condiment 6 Alert fatigue 7 The empty toolbox 8 Off-hour deployments 9 Wasting a perfectly good incident 10 Information hoarding: Only Brent knows 11 Culture by decree 12 Too many yardsticks

As requirements engineering continues to be recognized as the key to on-time and on-budget delivery of software and systems projects, many engineering programs have made requirements engineering mandatory in their curriculum. In

In addition, the wealth of new software tools that have recently emerged is empowering practicing engineers to improve their requirements engineering habits. However, these tools are not easy to use without appropriate training. Filling this need, *Requirements Engineering for Software and Systems, Second Edition* has been vastly updated and expanded to include about 30 percent new material. In addition to new exercises and updated references in every chapter, this edition updates all chapters with the latest applied research and industry practices. It also presents new material derived from the experiences of professors who have used the text in their classrooms.

Improvements to this edition include:

- An expanded introductory chapter with extensive discussions on requirements analysis, agreement, and consolidation
- An expanded chapter on requirements engineering for Agile methodologies
- An expanded chapter on formal methods with new examples
- An expanded section on requirements traceability
- An updated and expanded section on requirements engineering tools
- New exercises including ones suitable for research projects

Following in the footsteps of its bestselling predecessor, the text illustrates key ideas associated with requirements engineering using extensive case studies and three common example systems: an airline baggage handling system, a point-of-sale system for a large pet store chain, and a system for

a smart home. This edition also includes an example of a wet well pumping system for a wastewater treatment station. With a focus on software-intensive systems, but highly applicable to non-software systems, this text provides a probing and comprehensive review of recent developments in requirements engineering in high integrity systems. Although the use of data mining for security and malware detection is quickly on the rise, most books on the subject provide high-level theoretical discussions to the near exclusion of the practical aspects. Breaking the mold, *Data Mining Tools for Malware Detection* provides a step-by-step breakdown of how to develop data mining tools for malware d

This is the first book that presents a comprehensive overview of sustainability aspects in software engineering. Its format follows the structure of the SWEBOK and covers the key areas involved in the incorporation of green aspects in software engineering, encompassing topics from requirement elicitation to quality assurance and maintenance, while also considering professional practices and economic aspects. The book consists of thirteen chapters, which are structured in five parts. First the “Introduction” gives an overview of the primary general concepts related to Green IT, discussing what Green in Software Engineering is and how it differs from Green by Software Engineering. Next

“Environments, Processes and Construction”

presents green software development environments, green software engineering processes and green software construction in general. The third part, “Economic and Other Qualities,” details models for measuring how well software supports green software engineering techniques and for performing trade-off analyses between alternative green practices from an economic perspective. “Software Development Process” then details techniques for incorporating green aspects at various stages of software development, including requirements engineering, design, testing, and maintenance. In closing, “Practical Issues” addresses the repercussions of green software engineering on decision-making, stakeholder participation and innovation management. The audience for this book includes software engineering researchers in academia and industry seeking to understand the challenges and impact of green aspects in software engineering, as well as practitioners interested in learning about the state of the art in Green in Software Engineering.

Taking a learn-by-doing approach, Software Engineering Design: Theory and Practice uses examples, review questions, chapter exercises, and case study assignments to provide students and practitioners with the understanding required to design complex software systems. Explaining the

concepts that are immediately relevant to software designers, it begins with a review of software design fundamentals. The text presents a formal top-down design process that consists of several design activities with varied levels of detail, including the macro-, micro-, and construction-design levels. As part of the top-down approach, it provides in-depth coverage of applied architectural, creational, structural, and behavioral design patterns. For each design issue covered, it includes a step-by-step breakdown of the execution of the design solution, along with an evaluation, discussion, and justification for using that particular solution. The book outlines industry-proven software design practices for leading large-scale software design efforts, developing reusable and high-quality software systems, and producing technical and customer-driven design documentation. It also:

- Offers one-stop guidance for mastering the Software Design & Construction sections of the official Software Engineering Body of Knowledge (SWEBOK®)
- Details a collection of standards and guidelines for structuring high-quality code
- Describes techniques for analyzing and evaluating the quality of software designs

Collectively, the text supplies comprehensive coverage of the software design concepts students will need to succeed as professional design leaders. The section on engineering leadership for software designers covers the necessary ethical and

leadership skills required of software developers in the public domain. The section on creating software design documents (SDD) familiarizes students with the software design notations, structural descriptions, and behavioral models required for SDDs. Course notes, exercises with answers, online resources, and an instructor's manual are available upon qualified course adoption. Instructors can contact the author about these resources via the author's website:

<http://softwareengineeringdesign.com/>

SCM practices are recognised as core functional areas in assisting a project team to identify, control, audit, and report on all configuration items of a project. Consequently they are then better able to control changes to the working environment. Moreira presents a totally unique book, offering a “how-to” guide for SCM implementation for commercial and technology fields. A thoroughly practical approach; this guide includes examples and instruction of SCM tasks. This book has an easy to follow set of tasks that can be customized to assist a SCM professional in implementing SCM in a more efficient and expedient manner while also imparting SCM knowledge. Provides a customisable step-by-step process in implementing SCM Discusses typical SCM activities at project level and includes source control, change control, problem management, etc. An accompanying website contains templates,

procedures and other materials to aid understanding and encourage the practical applications of the material discussed throughout

www.wiley.com/go/moreira_software/ Anyone who has to implement SCM in his/her company at every level will need this book and find its practical approach useful

Emphasizing leadership principles and practices, *Antipatterns: Managing Software Organizations and People, Second Edition* catalogs 49 business practices that are often precursors to failure. This updated edition of a bestseller not only illustrates bad management approaches, but also covers the bad work environments and cultural traits commonly fou

Includes articles in topic areas such as autonomic computing, operating system architectures, and open source software technologies and applications.

This book discusses smart, agile software development methods and their applications for enterprise crisis management, presenting a systematic approach that promotes agility and crisis management in software engineering. The key finding is that these crises are caused by both technology-based and human-related factors. Being mission-critical, human-related issues are often neglected. To manage the crises, the book suggests an efficient agile methodology including a set of models, methods, patterns, practices and tools.

Together, these make a survival toolkit for large-scale software development in crises. Further, the book analyses lifecycles and methodologies focusing on their

impact on the project timeline and budget, and incorporates a set of industry-based patterns, practices and case studies, combining academic concepts and practices of software engineering.

Emphasizing leadership principles and practices, *Antipatterns: Managing Software Organizations and People, Second Edition* catalogs 49 business practices that are often precursors to failure. This updated edition of a bestseller not only illustrates bad management approaches, but also covers the bad work environments and cultural traits commonly fou.

"This book captures an in-depth knowledge base on the most current and useful concepts, applications, and processes relevant to the successful management of knowledge assets"--Provided by publisher.

This book compiles a number of contributions originating from the KESE (Knowledge Engineering and Software Engineering) workshop series from 2005 to 2015. The idea behind the series was the realignment of the knowledge engineering discipline and its strong relation to software engineering, as well as to the classical aspects of artificial intelligence research. The book introduces symbiotic work combining these disciplines, such as aspect-oriented and agile engineering, using anti-patterns, and system refinement. Furthermore, it presents successful applications from different areas that were created by combining techniques from both areas. *AntiPatterns: Identification, Refactoring, and Management* catalogs 48 bad management practices and environments common to software development, IT, and other organizations. The authors cover antipatterns

of management, along with environmental/cultural antipatterns and personality antipatterns/phenotypes. Through the classification of these Everything you need to anticipate and avoid the major pitfalls of project management-and ensure success! AntiPatterns in Project Management From the authors of the bestselling series on AntiPatterns come twenty new proven and practical solutions for successfully managing your software development project. in their familiar and entertaining style, the authors explore every phase of software development and identify the complex interaction of people, technology, and process issues that can cause a software development project to fail. With their pull-no-punches approach to project management, you'll learn how to balance these issues and ensure software development success. This book arms you with:

- * 6 People AntiPatterns that provide proven solutions to the people problems that occur within software development projects
- * 6 Technology AntiPatterns that help you deal with poor architectures and designs, and unstable technologies
- * 6 Processes AntiPatterns that help you uncover and correct the bad processes that ultimately cause a project to fail
- * The Standards AntiPattern that covers how to avoid the pitfalls and gain the benefits of adopting software development standards
- * The Collision AntiPattern that helps you successfully identify and eliminate multiple AntiPatterns that plague software development projects across different life-cycle phases, involving the disciplines of project management, software configuration management, and software development

Acces PDF Antipatterns Managing Software
Organizations And People Second Edition
Applied Software Engineering Series
[Copyright: 60f52514ae9d6d689ea910329b6e8447](#)